dCache 7.2

Dmitry Litvintsev

FIFE meeting

11/11/21

dCache releases

- We have been running dCache 5.2 since July 2020.
- Decided to jump one major release series to freshly released, bleeding edge golden release 7.2. To take advantage of:
  - Improved overall SciToken and JWT support.
  - SciToken (and JWT) support in xrootd.
  - Improved efficiency of HSM interaction of srm-bring-online by using request scheduler that takes into account file tape locations.
  - Progress on QoS and Data Lifecycle management.
  - **User and group quotas.**
- Took some hit for being too bold – had to fix a few bugs here and there. The kind of bugs that are seen under load and thus hard to test.

dCache and tokens

- dCache supports SiTokens and JWT in WebDAV and XRootD

```
# htgettoken -a fermicloud543.fnal.gov -i dune
Attempting to get token from https://fermicloud543.fnal.gov:8200 ... succeeded
Storing bearer token in /run/user/8637/bt_u8637
# httokendecode
{
     "wlcg.ver": "1.0",
     "aud": "https://wlcg.cern.ch/jwt/v1/any",
     "sub": "litvinse@fnal.gov",
     "nbf": 1636581332,
     "scope": "storage.create:/dune/scratch/users/litvinse compute.create
compute.read compute.cancel compute.modify storage.read:/dune",
     "iss": "https://cilogon.org/dune",
     "exp": 1636584937,
     "iat": 1636581337,
     "wlcg.groups": [
       "/dune"
     ],
     "jti":
"https://cilogon.org/oauth2/6faade280967b2ef432908921600d99f?type=accessToken&ts=
1636581337850&version=v2.0&lifetime=3600000"
}
```

dCache and tokens

- dCache maps “iss” claim to a group (VO) user:
  - `https://cilogon.org/dune -> 50381:9010`
- And allows that user to write into storage.create “scope” claim
  - `storage.create:/dune/scratch/users/<user>`

  with ownership of files created under that directory inherited from the ownership of that directory. Thus keeping uniform ownership across different protocols. And playing well with individual user storage accounting and Unix permissions.
- Different users, having different storage.create ”scope” claims cannot write into each other’s storage.create scopes.
- User 50381:9010 has read access to storage.read scope
  - `storage.read:/dune`

  regardless of Unix permissions on underlying entries.

XRootD

- Requires xroot-client >= 5.1 (personally tested with xrootd-client-5.3.)
- ”token enabled” door runs on a separate port – 1097 because currently we cannot mix different authn/authz shemes (GSI, unix, tokens) in one XRootD door.
- Example (my UID=8637)

```
# TOKEN=`cat /run/user/8637/bt_u8637`
# xrdcp foo xroot://fndca1.fnal.gov:1097/pnfs/fnal.gov/usr/dune/scratch/users/\
litvinse/xrootd_scitoken?authz=Bearer%20"${TOKEN}"
[799MB/799MB][100%][==================================================][99.87MB/s]
# echo $?
# 0

# ls -al/pnfs/fnal.gov/usr/dune/scratch/users/litvinse/xrootd_scitoken
-rw-r--r-- 1 8637 9010 837787900 Nov  9 18:45
/pnfs/fnal.gov/usr/dune/scratch/users/litvinse/xrootd_scitoken
```

WebDAV

- Curl works out of the box:

```
# TOKEN=`cat /run/user/8637/bt_u8637`
# curl -k -L -H "Authorization: Bearer ${TOKEN}" -Tjunk  \
https://fndca1.fnal.gov:2880/dune/scratch/users/litvinse/webdav_scitoken
837787900 bytes uploaded

# ls -al /pnfs/fnal.gov/usr/dune/scratch/users/litvinse/webdav_scitoken
-rw-r--r-- 1 8637 9010 837787900 Nov 10 16:42
/pnfs/fnal.gov/usr/dune/scratch/users/litvinse/webdav_scitoken
```

- Mind the relative path used (unlike in xrootd case):
  - /dune/scratch/users/<user>
- (one of the things to do on our plate – harmonize path handling)

User / Group quotas in dCache

- It has been a long standing request to implement user and group quotas so that:
  - Experiments/VOs can manage user quota within resources allocated to experiments.
  - dCache admins can manage resources allocated to Experiments/VOs by utilizing group quotas without resorting to creating multiple pool groups per each Experiment/VO.
- The main challenge of implementing quotas is having to balance the performance of individual file system operations with necessity to perform intensive data volume aggregations.
- An approach was chosen that involves namespace database scans on the back-end to *eventually* produce aggregated usage numbers that can be checked against pre-defined quotas on uploads.
- Initially Public dCache is configured to run these scans once a day.

User / Group quotas in dCache: nomenclature

- We support user (uid based) and group (gid based) quotas on:
  - REPLICA data (aka disk-only or ”persistent”).
  - CUSTODIAL data (tape-resident data).
- The CUSTODIAL data that is cached on disk is not subject to quota and behaves as usual – LRU eviction to free up to accept staged in and new data.
- The data in dCache scratch area is technically CUSTODIAL data with a special flag that tells the system – “do not store on tape, garbage collect”.
  - We could, in theory, impose user quotas on data in scratch as long as they do not also have tape-resident CUSTODIAL data.
- In the future we plan to assign OUTPUT class to scratch data and treat it separately.

User / Group quotas in dCache: behavior

- Due to periodic collection of aggregated data volumes by UID and GID:
  - Exceeding quota does not take effect till next scan. You still will be able to write until the scan completes on the back-end.
  - Conversely, removing excess data to go under quota does not take effect immediately. You won't be able to write until scan completes on the back-end.

User / Group quota in dCache: REST API

- Check out swagger UI (https://fndca.fnal.gov:3880/api/v1/)

User / Group quota in dCache: curling

- In all examples below “curl” means:

```
  # curl -L --capath /etc/grid-security/certificates  --cert /tmp/x509up_u`id -u` --cacert /tmp/x509up_u`id -u` --key  /tmp/x509up_u`id -u`
```

- Get user quota of all users (only I have quota at the moment):

```
  # curl -X GET "https://fndca.fnal.gov:3880/api/v1/quota/user" -H "accept: application/json"
[{"id":8637,"type":"USER","custodial":6442256574399,"replica":141504475421,"custodialLimit":109951162777600,"replicaLimit":109951162777600}]
```

- List only my user quota:

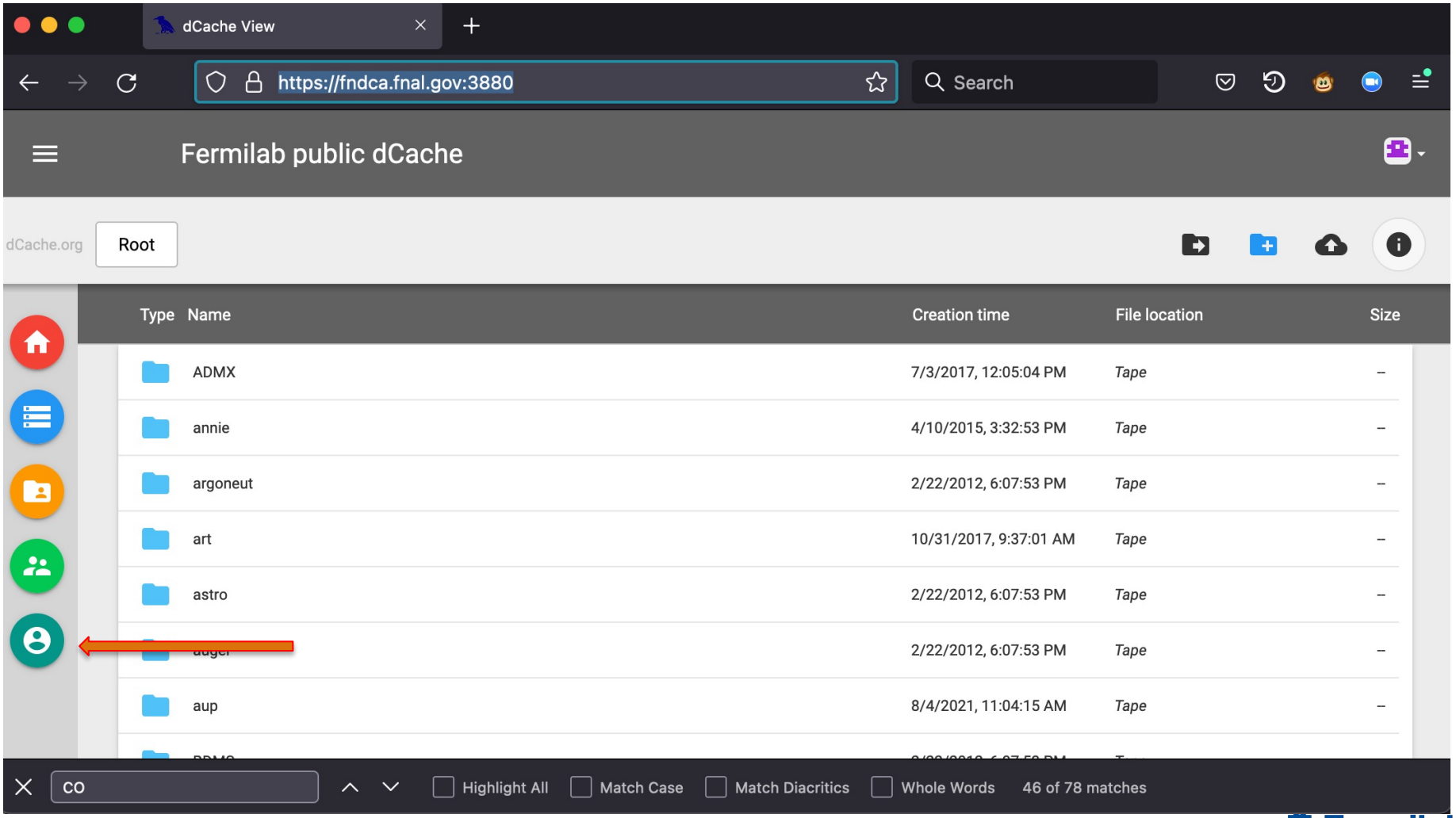
```
  # curl -X GET "https://fndca.fnal.gov:3880/api/v1/quota/user?user=true" -H "accept: application/json"
[{"id":8637,"type":"USER","custodial":6442256574399,"replica":141504475421,"custodialLimit":109951162777600,"replicaLimit":109951162777600}]
```

- List only my group quota (no group quota set):

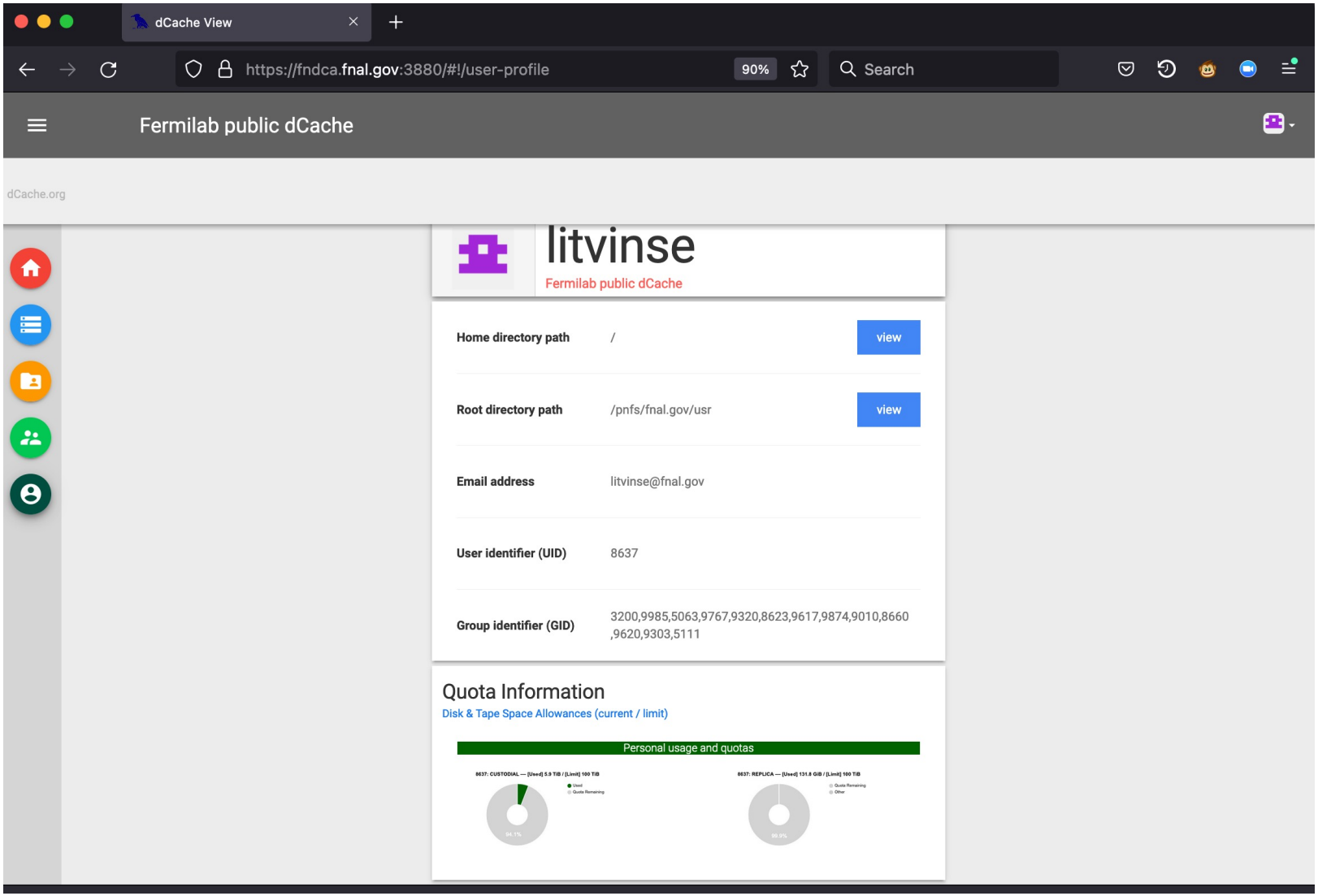
```
  # curl -X GET "https://fndca.fnal.gov:3880/api/v1/quota/group?user=true" -H "accept: application/json"
{"errors":[{"message":"No such quota exists","status":"404"}]}
```

User / group quota in dCache: user portal

- https://fndca.fnal.gov:3880/



User / group quota in dCache: user profile



Fermilab

Conclusion

- Updated dCache to the most recent golden release series - 7.2
- Addressed post-upgrade issues. Seem to be running smoothly now.
- Added token support in XRootD
- Sorted out data ownership and access issues when using tokens
- Rolled out User and Group quotas – looking for volunteers to use it. It is particularly helpful to manage “persistent” dCache areas.